Decision Matrix Decorrelation

Decision matrix systems often use a diagonal matrix D whose entries are selected from a small set such as {−1,1} or {0,1}. The matrix acts as a routing, reflection, masking, or sign-selection operator inside a larger transform such as:

$y = HD(x)Hx$

or more generally:

$y = AD(x)Bx$

where A and B are mixing transforms.

These systems can produce surprisingly rich nonlinear behavior, especially when iterated. However, an important practical issue appears when the entries of D are generated too directly from the coordinates of the vector x being processed.

If coordinate x_i directly determines diagonal entry D_{ii}, the system tends to collapse toward coordinate-wise nonlinearities similar to absolute value functions, clipping, or thresholding. Instead of distributed computation, each coordinate effectively “judges itself.” The resulting geometry becomes strongly axis aligned and loses much of the richness expected from distributed transforms.

The problem becomes especially visible in systems using sign gating:

$D_{ii}(x) = \text{sign}(x_i)$

because the transform immediately develops abs()-like folding behavior.

The central design goal is therefore decorrelation:

> the information used to decide the entries of D should not be strongly aligned with the coordinates being directly modulated by D.

Several useful decorrelation techniques exist.

Orthogonal mixing before gating is one of the simplest and most effective approaches. Instead of generating D directly from x, the vector is first transformed into another basis using a Walsh-Hadamard transform, FFT-like transform, or random orthogonal matrix. The decisions are then made from the mixed representation rather than the original coordinates. Because each transformed coordinate depends on many original coordinates, no single variable dominates its own gate.

A common construction is:

$u = Hx$

followed by generating D from u instead of x.

Permutation and derangement methods provide another simple mechanism. A coordinate can be gated by another coordinate selected through a permutation $\pi(i)$ where $\pi(i) \neq i$. This prevents direct self-gating and can significantly reduce trivial correlations. Random or quasirandom permutations are usually preferable to simple cyclic shifts because they better disrupt structured signal patterns.

Random projection hashing generalizes this further. Each decision bit is generated from a random hyperplane test:

$b_i = \text{sign}(r_i^T x)$

where r_i is a random vector. This creates distributed decision boundaries rather than coordinate-aligned thresholds. Nearby vectors tend to produce similar decisions while distant vectors decorrelate naturally. This approach is closely related to locality sensitive hashing and random feature methods.

Parity and Walsh-style constructions are also useful. Instead of one coordinate determining one gate, a gate can depend on the parity or sign product of several coordinates. This creates higher-order interactions and reduces separable behavior. In binary or ±1 systems these methods naturally connect to Walsh functions and Boolean Fourier analysis.

An especially interesting approach for small and medium systems is Zobrist-style hashing. Here each coordinate contributes a random symbolic value to the generation of the decision bits. The final gate vector becomes a distributed randomized fingerprint of the input rather than a direct measurement of any coordinate. Zobrist methods are sparse, inexpensive, incrementally updateable, and highly effective at breaking low-order correlations.

Weyl sequence **permutations** provide another elegant decorrelation mechanism. A Weyl sequence:

$s_t = (s_0 + t\omega) \bmod 2^n$

with odd additive constant ω generates a full permutation cycle over the integer ring modulo 2n. Well chosen Weyl constants avoid repeated low-order alignment with the original coordinate ordering and distribute interactions evenly across bit planes. Constants derived from irrational ratios such as the golden ratio are especially effective because they minimize resonance and hidden periodic structure.

This matters because structured iterative systems can otherwise develop invariant subspaces, lattice artifacts, or repeated cycles caused by stable coordinate alignments. Quasirandom permutations and Weyl indexing help destroy these resonances while remaining computationally cheap.

Soft decision matrices provide another important extension. Instead of hard binary gates, the diagonal entries can be continuous functions such as tanh(), signed square roots, or clipped linear ramps. This reduces sharp folding and allows smoother state-space dynamics while preserving distributed routing behavior.

The broader geometric distinction is between local gating and distributed gating. Local gating produces coordinate-wise nonlinearities and axis-aligned partitions. Distributed gating produces high-dimensional hyperplane partitions and global symbolic interactions. The latter behaves more like scattering, randomized routing, associative memory, or hyperdimensional symbolic computation than a conventional neural activation function.

.